CurVe – a LaTeX 2ε class package for making **Cur**ricula **V**ita**e**. [*]

Didier Verna
mailto:didier@lrde.epita.fr
http://www.lrde.epita.fr/~didier

September 14, 2005

Abstract

CurVe provides a LaTeX 2ε class that hopefully will make your life easier when you want to write your CV. It provides you with a set of commands to create rubrics, entries in these rubrics etc. CurVe will then properly format your CV for you (possibly splitting it onto multiple pages), which is usually the most painful part of CV writing. Another nice feature of CurVe is its ability to manage different CV "flavors" simultaneously. It is in fact often the case that you want to maintain slightly divergent versions of your CV at the same time, in order to emphasize on different aspects of your background.

1 Getting CurVe

CurVe can be obtained from any CTAN archive, in the `macros/latex/contrib` subdirectory. You can also download it directly from my website (online documentation available there), at the URL above. Please follow the links on the left menu.

If you are a Debian unstable user (unstable referring to Debian, not you), unofficial source and i386 packages are available (thanks to Geoffroy Fouquier for providing this facility). The package name is `curve`. Here's the `source.list` entry to use:

```
deb http://www.lrde.epita.fr/debian/ unstable/i386/
deb-src http://www.lrde.epita.fr/debian/ sid/source/
```

For installation instructions, please read the `README` file included in the distribution.

2 Overview

The CurVe package provides you with a document class for writing curricula vitae. The primary purpose of this package is to offer a set of predefined commands to

[*] This document describes CurVe v1.10, release date 2005/09/14.

specify the contents of your CV, while removing from you the burden of formatting it. This has two important consequence however: CurVe will impose that you conform to its document structuring scheme, and will expect that you like the way it formats things :-). If you prefer another structure for your CV, or if you don't like the formatting (although it is highly configurable), then CurVe is probably not for you.

Once you have installed CurVe, you might want to start with processing the example file `cv.tex`. This will give you an idea of what a non customized CV looks like with CurVe. You can also throw an eye to my own CV, which is written with CurVe and has some more fancy hackery on top of it. It's in French, but only the appearance is important for you. . . My CV can be found at `http://www.lrde.epita.fr/~didier/perso/cv.php`.

2.1 Document Layout

A CurVe CV begins with two optional headers (upper left and upper right) in which you usually put your name, address, email, whether you're married and so on. These headers will respectively be left and right aligned. As of version 1.4, CurVe lets you insert a small identity photo in the headers, either on the left, on the right, or between them. After these headers come an optional title and/or subtitle, which will be centered on the page.

2.1.1 Rubrics

The remaining of the document is composed of sections called "rubrics" in the CurVe terminology. A rubric represents a major topic that you want to detail in your CV. Typical rubrics are "Education", "Professional Experience" and the like. Rubrics have a title (centered by default) and appear under the form of properly aligned "entries" (see below). If a rubric has to be split across different pages, its title will be repeated automatically.

2.1.2 Entries

An entry is an item of information related to the rubric under which it appears. An entry has a "contents", and an optional "key" under which it is classified. For instance, under the "Education" rubric, you could state that you got a Ph.D. in computer science in the year 2000. In that case, the year would be the entry's key, and the "Ph.D. in computer science" part would be the entry's contents. CurVe aligns both keys and contents together. Keys are optional in order for you to classify several entries together (without repeating the same key over and over again).

2.1.3 Subrubrics

Additionally, you might want to further split your rubrics into "subrubrics". For instance, in my own CV, I have a "Professional Experience" rubric, with three subrubrics: "Teaching", "Research" and "Development". This can be accomplished with a special command. Subrubrics are displayed in alignment with the entries' contents by default, but are formatted differently so that they remain distinguishable.

2.2 Document Structure

2.2.1 Source File Splitting

CurVe is based on the `LTXtable` package by David Carlisle. I won't go into gory details, but this has an important implication: **each rubric must be in its own separate file**. In other words, your CV's main source file is really a skeleton whose major task is to include the different rubrics from their respective source files.

This is not much of a hassle, really, and it actually made my life easier when I implemented the "flavor" mechanism described below.

2.2.2 The "flavor" Mechanism

It is often desirable to maintain several slightly divergent versions of one's CV at the same time. For instance, when I was looking for a job some time ago, I had a version of my CV emphasizing on Artificial Intelligence, and another emphasizing on Distributed Virtual Reality. Only the title and some entries in the "Professional Experience" rubric were a bit different; the main skeleton basically remained the same.

CurVe provides an easy-to-use mechanism for maintaining different "flavors" of your CV at the same time. You basically write different versions of (some of) your rubrics in different files, tell CurVe which flavor you want to format (CurVe can even ask you which one to use directly) and that's it. CurVe will use the global skeleton, and whenever it finds a rubric file specialized for that particular flavor, it will use it. Otherwise, it will simply fall back to the default one (no particular flavor).

3 Using CurVe

First of all, please note that the `ltxtable` and `calc` packages are required. If you're using the identity photo feature, the `graphicx` package is also needed. You don't have to load them explicitly though. As long as LaTeX 2$_\varepsilon$ can locate them, they will be used automatically.

3.1 Writing the Skeleton File

Say `\documentclass[`⟨*options*⟩`]{curve}` at the beginning of your skeleton file in order to use CurVe. The available options are described along the text, where appropriate.

3.1.1 Making Headers

`\leftheader` `\rightheader` The `\leftheader` and `\rightheader` macros take one mandatory argument which defines respectively the contents of the upper left and upper right headers. They can be used in the document's preamble only. The headers will respectively be flushed to the left and to the right.

`\photo` If you want to insert a small identity photo into the header part, you can use the `\photo` macro (available since version 1.4). It takes a mandatory argument in which you pass the image file name, as you would to `\includegraphics`. This macro also takes an optional argument which lets you specify the horizontal position of the photo: the values can be `l` (the default), `c` or `r` meaning that the photo will appear on the left, center, or right.

`\photoscale`
`\photosep`
`\headerscale`

The headers' horizontal layout is further controled by three additional macros. The `\photoscale` macro specifies the amount of text width that the photo should occupy. This should be a number between 0 and 1. By default, 0.1 is used (meaning 10% of `\textwidth`). The `\photosep` macro is a LaTeX length that specifies the space to leave between the side of the photo and the next headers's text. This is used only when the photo is on the left or right. By default, `10pt` is used. Finally, `\headerscale` specifies the proportion of the *remaining* space that the *left* textual header should occupy. It works like `\photoscale` and amounts to 0.5 by default.

Let me take an example to make this clearer. Suppose you have a `\photoscale` of 0.1 and a `\photosep` of `10pt`. The *remaining* space, that is, the space occupied by the textual headers, amounts to 90% of the text width, minus 10 points. If you then specify a `\headerscale` of 0.6, then the left header will take 60% of that remaining space, and the right one the other 40%.

`\headerspace`

`\headerspace` is the amount of extra vertical space to put after the headers. This is a LaTeX length that defaults to `10pt`.

`\makeheaders`

If you have defined headers, make them appear by calling `\makeheaders` just after the beginning of your document. Note that calling this macro assumes that you have previously defined both headers (possibly empty, though). Otherwise, an error will be signaled. As of version 1.4, the `\makeheaders` command accepts an optional argument that controls the vertical alignment. When given, this argument must be either `t` (for top), `b` (for bottom) or `c` (for center; the default).

3.1.2 Making Titles

`\title`
`\subtitle`

The `\title` and `\subtitle` macros take one mandatory argument which define respectively your CV's title and subtitle. They can be used in the document's preamble only. These titles will be centered on the page.

`\titlespace`

`\titlespace` is the amount of extra vertical space to put after the title(s). This is a LaTeX length that defaults to `0pt`.

`\titlefont`
`\subtitlefont`

The `\titlefont` and `\subtitlefont` macros take one mandatory argument which redefine the fonts to use for the title and the subtitle. They can be used in the document's preamble only. By default, `\Huge\bfseries` and `\Huge\itshape` are used respectively.

`\maketitle`

If you have defined a title (and possibly a subtitle), make it (them) appear by calling `\maketitle` after the beginning of your document, and just after `\makeheaders` if you happen use it. It is possible to omit the subtitle, but if you call `\maketitle` without having defined at least a title, an error will be signaled.

3.1.3 Choosing a Flavor

As you already know, each rubric must reside in its own separate file. For instance, if you have a "Professional Experience" rubric, you would write its contents into a file named `experience.tex`. The flavor mechanism works by assigning a pre-extension to rubric file names. For instance, suppose you want to make a special flavor of your CV emphasizing on "distributed virtual reality". You would call this flavor “dvr”, and write the modified “Professional Experience” rubric into a file named `experience.dvr.tex`.

`\flavor` The `\flavor` macro takes one mandatory argument which specifies the flavor to use (in our example, `dvr`). Although this might be of little use, it is possible to change the flavor anywhere, even right in the middle of your CV’s skeleton.

`ask` Instead of using the `\flavor` macro, you can make CurVe ask you at run-time which flavor to use by passing the `ask` option to it.

3.1.4 Including Rubrics

Apart from making headers and titles, the body of your skeleton file will usually contain nothing but directives to include the different rubrics of your CV.

`\makerubric` To include a rubric in your document, use `\makerubric`. This macro takes one mandatory argument which specifies the rubric to include at that point. The argument actually corresponds to the rubric file name **without any extension**. Continuing our previous example, you would say `\makerubric{experience}`. First, CurVe will try to find such a rubric file specific for the current flavor in use, (e.g. `experience.dvr.tex`). If that fails, it will fall back to a non-flavored file (here, `experience.tex`). This allows you to specialize only the required rubrics and use the default ones otherwise.

3.2 Writing a Rubric File

3.2.1 The `rubric` Environment

`rubric` The whole contents of a rubric file must be enclosed in a `rubric` environment. This environment takes one mandatory argument which specifies the rubric’s title. When a rubric crosses several pages, its title is restated with a “continuation” text appended.

`\rubricalignment` As of version 1.6, the rubric titles horizontal alignment can be changed thanks to the `\rubricalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right relative to the whole text width), and `cl` and `cc` (meaning left or centered relative to the entries’contents). By default, rubric titles are centered (`c`).

`\rubricfont` The `\rubricfont` macro takes one mandatory argument which redefines the font to use for rubric titles. By default, `\Large\bfseries` is used.

`\rubricspace` `\rubricspace` is the amount of extra vertical space to put after the rubric title. This is a LaTeX length that defaults to `10pt`.

3.2.2 Making Rubric Entries

`\entry` You create entries in your rubrics by calling the `\entry` macro. The first (optional) argument specifies the key, and the second (mandatory) one specifies the contents. Both keys and contents are aligned within each rubric.

`\entry*` Actually, the `\entry` macro was somewhat ill-designed at the first place. The `rubric` environment pretty much behaves as an `itemize` one, hence the idea of using an `\item`-like syntax. As of version 1.2, CurVe provides an `\entry*` macro which behaves like `\item` in lists: it takes the same first optional argument as the non starred version, but has no other argument. The entry’s contents simply consists of the text following the macro call, up to the next `\entry`, `\entry*` or `\subrubric` (see below) call.

`\keyalignment`

As of version 1.7, entries'keys horizontal alignment can be changed thanks to the `\keyalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right). By default, keys are left aligned (`l`).

`\keyfont`

The `\keyfont` macro takes one mandatory argument which redefines the font to use for the entries' keys. By default, the standard document font is used.

`\prefix`

Each entry's contents can be prefixed with a visual clue (a symbol for instance). This comes in handy to make a clear distinction between different entries sharing the same key (which is not repeated). The `\prefix` macro takes one mandatory argument which redefines the prefix to use. By default, `\textbullet` is used.

`skipsamekey`

While maintaining your CV, you might end up reorganizing your entries and even get entries with the same key. Normally, CurVe blindly prints the keys regardless of their values. If you don't want repetition, you would have to remove keys by hand which can be cumbersome. As of version 1.10, CurVe can skip all but the first of a series of identical keys automatically, provided that you use the `skipsamekey` option.

3.2.3 Making Subrubrics

`\subrubric`

Within a single rubric, you can further separate entries into subrubrics. In order to do this, the `\subrubric` macro is provided. Its mandatory argument specifies the subrubric's title.

`\subrubricalignment`

As of version 1.6, the subrubrics horizontal alignment can be changed thanks to the `\subrubricalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right relative to the whole text width), and `cl` and `cc` (meaning left or centered relative to the entries'contents). By default, subrubrics are left-aligned with the entries' contents (`cl`).

`\subrubricfont`

The `\subrubricfont` macro takes one mandatory argument which redefines the font to use for the subrubrics. By default, `\Large\itshape` is used.

`\subrubricspace`
`\subrubricbeforespace`

`\subrubricspace` controls the amount of extra vertical space to put after subrubrics. This is a LaTeX length that defaults to `5pt`. `\subrubricbeforespace` controls the amount of extra vertical space to put *before* a subrubric when there are entries above. This is a LaTeX length that defaults to `10pt`.

3.3 Standard Class Features

3.3.1 Page Size and layout

`a4paper`
`a5paper`
`b4paper`
`letterpaper`
`legalpaper`
`executivepaper`
`landscape`
`oneside`
`twoside`

The `a4`, `a5`, `b4`, `letter`, `legal` and `executive` “paper” options allow you to select the type of page format you want. By default, `letterpaper` is used. The `landscape` options switches the horizontal and vertical settings. I'm not sure why I propose this option. Nobody wants to write a CV in landscape mode, right ?

As of version 1.6, CurVe also supports the standard `oneside` and `twoside` class options. By default, `oneside` is used. In `twoside` mode, odd and even pages have a different geometry and headings layout.

3.3.2 Font Size

`10pt`
`11pt`
`12pt`

The `10pt`, `11pt` and `12pt` options let you choose the size of the default font you want to use. By default, `10pt` is used.

3.3.3 Output Mode

`final`
`draft`

In `draft` mode, a black rule will be drawn at the end of overfull lines (as done by standard classes). Due to CurVe using the `LTXtable` package, a call to `\setlongtables` is performed in `final` mode. Please refer to the next section for more information on this. By default, `final` is used.

3.3.4 Page styles

As of version 1.6, CurVe supports the standard LaTeX page style mechanism. Available styles are `empty`, `plain`, `headings` and `myheadings`. These styles have their usual meaning, given that rubric and subrubric names are used for marking purpose (the equivalent of chapters and sections in books for instance). By default, the page style is `empty`.

3.3.5 Bibliography

Most scientists include their own list of publications in their CV. The first thing you can do is create your own bibliography manually, and although this may appear boring, I actually encourage people to do so for at least three reasons (only my opinion of course):

- A CV should be strictly formatted and coherent in layout. Bibliography is no exception to this rule. In other words, it is prettier to have your publications formatted like the rest of your CV.
- Automatic bibliography generation tools produce references, which is silly in a CV because you don't actually reference your papers anywhere (or do you ?). So better to sort them another way, like, by year of publication as I do in my own CV.
- Manually adding, like, what ? Half a dozen papers a year in your CV is not that much of a burden after all.

`thebibliography`
`\bibitem`

Some people however have expressed the wish of having standard bibliography support in CurVe. Version 1.2 provides that. The standard `thebibliography` environment is now supported along with its `\bibitem` companion. The behavior is actually that of a `rubric` environment with its `\entry*` companion. This fact has two implications: firstly, the argument to the environment is unused in CurVe (but remains for compatibility with the rest of LaTeX) because CurVe itself formats the keys and contents properly aligned. Secondly, the bibliographic environment **must** reside in its own file, as any other rubric. Don't forget that if you happen to write the environment manually.

`\nocite`
`\bibliographystyle`
`\bibliography`

If you want to use BibTeX, that's also possible of course. Do it as you would do in a random paper. You will probably issue a `\nocite{*}` command followed by a call to `\bibliography`. In CurVe, this uses the `bbl` file as a rubric one.

3.3.6 Internationalization

`english`
`french, francais`
`spanish`
`italian`
`german, ngerman`
`danish`
`dutch`
`portuges, portuguese`
`brazilian, brazil`

CurVe currently supports English, French, Spanish, Italian, German, Danish, Dutch and Portuguese. You can select the language you want to use by using the corresponding option. The `french` and `francais` options are synonyms. The `german` and `ngerman` options are currently equivalent. So are the `portuges`, `portuguese`, `brazil` and `brazilian` options.

If you want a finer grain on the language-dependent parts of CurVe, the following macros are provided.

`\continuedname` The `\continuedname` macro takes one mandatory argument which redefines the continuation text output when rubrics extend across several pages. By default, "⟨*space*⟩(continued)" is used in English. Although this might be of little use, it is possible to change the continuation text in the middle of your document, provided that you do so outside the `rubric` environment.

`\listpubname` The `\listpubname` macro takes one mandatory argument which redefines the title of the bibliographic section (when you use the provided bibliography support). By default, "List of Publications" is used in English.

4 Hints, Tricks, Tips

Here are some tips that I use for my own CV. You might find them of some interest.

4.1 Page Geometry

First of all, it is common to have very thin margins in curricula vitae. CurVe does not do anything special about this because I don't think that belongs to its duty. The `geometry` package comes in handy if you want to reduce your margins.

4.2 The `ltx` Extension

Personally, I prefer to keep `.tex` for TeX files, and use the `ltx` extension for LaTeX. This is supported by CurVe which will actually prefer `ltx` files over `tex` ones, especially when including rubrics. To be more precise, suppose you are building a flavor `flv` of your CV. A call to `\makerubric{foo}` will try to use the following files in that order:

```
foo.flv.ltx
foo.flv.tex
foo.ltx
foo.tex
```

4.3 Longtables

The `LTXtable` package on which CurVe is based is a mix of `tabularx` and `longtables`. If you read the documentation of the later, you will discover that for table width computing reasons (especially when a table crosses several pages), LaTeX has to be called twice, sometimes three times, with the last run involving a call to `\setlongtables`.

Normally, you shouldn't have problems with CurVe because all tables are set to the maximum width. However, for safety reasons (I mean, just to be sure. . .), CurVe automatically calls `\setlongtables` in `final` mode. If you experiment problems with the formatting, you should process your document once or twice in `draft` mode, and a second or third time in `final` mode.

Ah, and also, since you're basically working in tabular environments, don't forget that you are not allowed to use the `\\` command. . .

4.4 Managing Different Flavors

If you maintain different flavors of your CV at the same time, you probably want to rebuild all of them after any modification. Since you have a single skeleton file for all of them (say, `cv.tex`), the output file will have the same name for all flavors (say, `cv.dvi`). This can bother you if you want all flavors of your formatted CV available at the same time.

To remedy this problem, I usually use the `ask` option and a makefile to build the different flavors and move the output file to flavor-specific name. Here is a typical makefile target that should clarify (or maybe darken ?) what I am saying:

```
cv.$(FLAVOR).dvi: cv.ltx $(RUBRICS)
	echo $(FLAVOR) | latex cv.ltx
	mv cv.dvi $@
```

As you can see, the shell is responsible for answering the question.

4.5 More On Flavors

In order to implement the flavor mechanism, the LaTeX macro `\input` has been redefined to look for "flavored" files first. This is actually very nice because you can use it if you want to make different flavors of text that does not belong in rubrics.

For instance, suppose you want a special version of the subtitle of your CV for the flavor `flv`. Create a file called `subtitle.flv.ltx` and put something like "`\subtitle{special subtitle}`" in it. Do something similar for the default subtitle. Now go to the skeleton of your CV, and write `\input{subtitle}` in the preamble. That's it. You'll have different subtitles in your different CV flavors.

5 AUC-TeX support

AUC-TeX is a powerful major mode for editing TeX documents in Emacs or XEmacs. In particular, it provides automatic completion of macro names once they are known. CurVe supports AUC-TeX by providing a style file named `curve.el` which contains AUC-TeX definitions for the relevant macros. This file should be installed to a location where AUC-TeX can find it (usually in a subdirectory of your LaTeX styles directory). Please refer to the AUC-TeX documentation for more information on this.

As of version 1.2, CurVe has an improved AUC-TeX support. Most notably, the command `M-Ret` will insert an `\entry*` macro within a `rubric` environment. Also, the `\makerubric` macro handling now removes both the file extension and the file flavor extension.

6 Changes

v1.10 Support automatic skipping of identical keys, suggested by Akim Demaille
Fix alignment problem with empty prefix, reported by Jonas Haulin

v1.9 Fix incompatibilities with the `bibentry` package, reported by Joris Desmet
Fix standard bibliography support (broken in v1.8)

v1.8 Prevent page breaks after subrubric headings

v1.7 Support for key horizontal alignment
\raggedleft and \raggedright can now be used within individual entries
Fix typo in Danish version of \continuedname

v1.6 Support for rubric and subrubric titles horizontal alignment
Support for standard LaTeX page style mechanism
Support for oneside and twoside options
Support for Portuguese thanks to Adiel Mittmann <adiel@inf.ufsc.br>
Fix bug in \bibliography: protect against non existant files, reported by Andrew Comport
Fix conflict with hyperref in some bibliography definitions

v1.5 Support for Dutch thanks to Thomas Delaet
<Thomas.Delaet@student.kuleuven.ac.be>
Fix typo in rubric environment, reported by Torsten Liesk

v1.4 Support for photo inclusion
Support for headers horizontal scaling
Optional argument to \makeheaders for vertical alignment, suggested by Dan Luecking

v1.3 Support for Danish thanks to Kim Rud Bille <krbi01@control.auc.dk>

v1.2 Support for standard bibliography mechanism(s)
New macro \entry*
Improvements in AUC-TeX support
Support for German thanks to Harald Harders <h.harders@tu-bs.de>
Support for Spanish thanks to Agustín Martín <agusmba@terra.es>

v1.1 Support for Italian thanks to Riccardo Murri <murri@phc.unipi.it>

7 The Code

First, the class announcement and the initial requirements:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{curve}[2005/09/14 v1.10
                      Curriculum Vitae class for LaTeX2e]

\RequirePackage{ltxtable}
\RequirePackage{calc}

```

The following macro tests strings equality. It avoids the hassle of this stupid TeX scheme that prevents simple conditionals imbrication.

```
\newif\ifstrok\strokfalse
\def\strtest#1#2{%
  \def\@strone{#1}\def\@strtwo{#2}%
  \ifstrok\else\ifx\@strone\@strtwo\stroktrue\fi\fi}

```

7.1 The Rubric File

We don't want to output an extra subrubricbeforespace if no entry is present before the subrubric. This is done by using an \@beforespace command which is set to 0pt at the beginning of each rubric, and switched to the proper value when an entry is added.

The @nextentry command is used to implement \entry* while maintaining backward compatibility with \entry and \subrubric. A new entry or a subrubric might have to close the preceding entry if it was opened using the starred form.

```
\gdef\@nextentry{}

```

7.1.1 Entries

\keyfont

```
\def\@keyfont{}
\newcommand\keyfont[1]{\gdef\@keyfont{#1}}

```

\keyalignment

```
\newcolumntype{k}{>{\@keyfont}l}
\newcommand\keyalignment[1]{%
  \strokfalse\strtest{#1}{l}\strtest{#1}{r}\strtest{#1}{c}%
  \ifstrok\else%
    \ClassError{curve}{Invalid key alignment}{%
      You have called \protect\keyalignment\space with an invalid value.%
      \MessageBreak
      Valid options include l, c, and r.\MessageBreak
      Type X <return> to quit, fix the typo, and rerun LaTeX.}%
  \fi
  \newcolumntype{k}{>{\@keyfont}#1}
}

```

\prefix

```
\def\@prefix{\textbullet}
\newcommand\prefix[1]{\gdef\@prefix{#1}}

```

\entry As of version 1.10, CurVe can skip keys identical to the previous one, if the option skipsamekey is used.

```
\def\@maybekey#1{%
  \def\@newkey{#1}%
  \ifx\@previouskey\@newkey\else%
    #1\gdef\@previouskey{#1}%
  \fi}

\def\@key#1{#1}
\DeclareOption{skipsamekey}{\let\@key\@maybekey}

\newcommand\@entry[2][]{%
  \gdef\@nextentry{}\egroup% end of \noalign opened in \entry.
  \@key{#1}&\@prefix&#2\\}

```

```
\newcommand\@sentry[1][]{%
  \gdef\@nextentry{\\}\egroup% end of \noalign opened in \entry.
  \@key{#1}&\@prefix&}

\newcommand\entry{%
  \@nextentry
  \noalign\bgroup\gdef\@beforespace{-\subrubricbeforespace}%
  \@ifstar{\@sentry}{\@entry}}

```

7.1.2 Subrubrics

\subrubricfont
\subrubricbeforespace
\subrubricspace

```
\def\@subrubricfont{\Large\itshape}
\newcommand\subrubricfont[1]{\gdef\@subrubricfont{#1}}

\newlength\subrubricbeforespace
\setlength\subrubricbeforespace{10pt}

\newlength\subrubricspace
\setlength\subrubricspace{5pt}

```

\subrubricalignment
\subrubric

Note that `\@subrubricmark` is called outside the raisebox. That's because otherwise, the mark would not go to the toplevel page vertical box, and TeX would not notice it.

```
\let\@subrubricmark\@gobble

\def\@@subrubric#1{%
  \parbox{0cm}{%
    \raisebox{\@beforespace}{\@subrubricfont#1}%
    \par\vspace\subrubricspace}%
  \@subrubricmark{#1}}

```

The normal intercolumn space between the prefix and the entry's content is replaced with an unbreakable space. This causes a problem (fixed in version 1.10) with the [sub]rubric alignments cl and cc when the prefix is empty, because the unbreakable space in question slightly shifts the entry's content to the right. If we want a proper alignment, we then have to take this offset into account in the cl and cc multicolumns, for both rubrics and subrubrics. The following macro implements this:

```

\def\@clcccolsep{\hspace{\tabcolsep}\ifx\@prefix\@empty~\fi}

\def\@subrubric@l#1{\multicolumn{3}{@{}l}{\@@subrubric{#1}}}
\def\@subrubric@c#1{\multicolumn{3}{@{}c}{\@@subrubric{#1}}}
\def\@subrubric@r#1{\multicolumn{3}{@{}r}{\@@subrubric{#1}}}
\def\@subrubric@cl#1{&\multicolumn{2}{@{\@clcccolsep}l}{\@@subrubric{#1}}}
\def\@subrubric@cc#1{&\multicolumn{2}{@{\@clcccolsep}c}{\@@subrubric{#1}}}

\let\@subrubric\@subrubric@cl

\newcommand\subrubricalignment[1]{%
```

```
  \def\@curve@temp@a{\let\@subrubric}
  \expandafter\@curve@temp@a\csname @subrubric@#1\endcsname
  \@ifundefined{@subrubric}{%
    \ClassError{curve}{Invalid subrubric alignment}{%
      You have called \protect\subrubricalignment\space with an invalid value.%
      \MessageBreak
      Valid options include l, c, r, cl and cc.\MessageBreak
      Type X <return> to quit, fix the typo, and rerun LaTeX.}}
}

\newcommand\subrubric[1]{%
  \@nextentry
  \noalign{\gdef\@nextentry{}}%
  \@subrubric{#1}\\*}

```

7.1.3 Rubrics

The `\@almosttextwidth` length remains only for backward compatibility. It is not used anymore.

`\rubricfont`
`\rubricspace`

```
\newlength{\@almosttextwidth}
\AtBeginDocument{\setlength\@almosttextwidth{\textwidth-\hfuzz}}

\def\@rubricfont{\Large\bfseries}
\newcommand\rubricfont[1]{\gdef\@rubricfont{#1}}

\newlength\rubricspace
\setlength\rubricspace{10pt}

```

`\rubricalignment`
`rubric`

```
\let\@rubricmark\@gobble

\def\@@rubrichead#1{\@rubricfont#1\par\vspace\rubricspace}

\def\@rubrichead@l#1{\multicolumn{3}{@{}l}{\@@rubrichead{#1}}\\}
\def\@rubrichead@c#1{\multicolumn{3}{@{}c}{\@@rubrichead{#1}}\\}
\def\@rubrichead@r#1{\multicolumn{3}{@{}r}{\@@rubrichead{#1}}\\}
\def\@rubrichead@cl#1{&\multicolumn{2}{@{\@clcccolsep}l}{\@@rubrichead{#1}}\\}
\def\@rubrichead@cc#1{&\multicolumn{2}{@{\@clcccolsep}c}{\@@rubrichead{#1}}\\}

\let\@rubrichead\@rubrichead@c

\newcommand\rubricalignment[1]{%
  \def\@curve@temp@a{\let\@rubrichead}
  \expandafter\@curve@temp@a\csname @rubrichead@#1\endcsname
  \@ifundefined{@rubrichead}{%
    \ClassError{curve}{Invalid rubric alignment}{%
      You have called \protect\rubricalignment\space with an invalid value.%
      \MessageBreak
      Valid options include l, c, r, cl and cc.\MessageBreak
      Type X <return> to quit, fix the typo, and rerun LaTeX.}}
}

```

Marking commands don't seem to work in longtable headings. So the rubric mark is issued just after it.

As of version 1.7, `\raggedleft` and `\raggedright` are redefined in order to work within individual entries. This redefinition simply consists in removing the `\\` definition since it's not available anyway, and also to remove the `\parskip` setting since it's `0pt` in the whole class.

```
\newenvironment{rubric}[1]{%
  %% \begin{rubric}
  \def\raggedright{%
    \@rightskip\@flushglue\rightskip\@rightskip\leftskip\z@skip}%
  \def\raggedleft{%
    \rightskip\z@skip\leftskip\@flushglue\parfillskip\z@skip}%
  \gdef\@beforespace{0pt}%
  \gdef\@nextentry{}%
  \gdef\@previouskey{}
  \begin{longtable}{@{}kl@{~}X}
    \@rubrichead{#1}
    \endfirsthead
    \@rubrichead{#1\@continuedname}
    \endhead
    \noalign{\@rubricmark{#1}}}{%
    %% \end{rubric}
    \@nextentry
  \end{longtable}}

```

`\continuedname`

```
\newcommand\continuedname[1]{\gdef\@continuedname{#1}}

```

7.2 The Skeleton File

7.2.1 Headers

`\headerscale`
`\headerspace`

```
\def\header@scale{.5}
\newcommand\headerscale[1]{\gdef\header@scale{#1}}
\@onlypreamble\headerscale

\newlength\headerspace
\setlength\headerspace{10pt}

```

`\leftheader`
`\rightheader`

If the user calls `\makeheaders` without specifying headers first, an error will be generated. The same applies for the title (not the subtitle), but this is already managed by LaTeX itself.

```
\def\@leftheader{%
  \ClassError{curve}{No \protect\leftheader\space given}{%
    You have called \protect\makeheaders, %
    but you didn't provide a left header.\MessageBreak
    Type X <return> to quit, add a call to \protect\lefheader\space %
    in the preamble of your CV,\MessageBreak
    and rerun LaTeX.}}
\newcommand\leftheader[1]{\gdef\@leftheader{#1}}
```

```
\@onlypreamble\leftheader

\def\@rightheader{%
  \ClassError{curve}{No \protect\rightheader\space given}{%
    You have called \protect\makeheaders, %
    but you didn't provide a right header.\MessageBreak
    Type X <return> to quit, add a call to \protect\rightheader\space %
    in the preamble of your CV,\MessageBreak
    and rerun LaTeX.}}
\newcommand\rightheader[1]{\gdef\@rightheader{#1}}
\@onlypreamble\rightheader

```

`\photoscale` `\photosep` `\photo`

```
\def\photo@scale{.1}
\newcommand\photoscale[1]{\gdef\photo@scale{#1}}
\@onlypreamble\photoscale

\newlength\photosep
\setlength\photosep{10pt}

\newcommand\photo[2][l]{%
  \RequirePackage{graphicx}
  \strokfalse\strtest{#1}{l}\strtest{#1}{r}\strtest{#1}{c}%
  \ifstrok\else\ClassError{curve}{Invalid argument to \protect\photo}{%
    Argument 2 of \protect\photo must be 'l', 'r' or 'c'.}\fi
  \def\tmp@cmd{\global\let\makeheaders@}
  \expandafter\tmp@cmd\csname makeheaders@#1\endcsname
  \gdef\photo@file{#2}}
\@onlypreamble\photo

```

`\makeheaders` These different versions of the photo inclusion command exist for proper alignment of the picture itself with the left and right headers.

```
\newlength\photo@width

\def\includephoto@t{%
  \raisebox{.7\baselineskip-\height}{%
    \includegraphics[width=\photo@width]{\photo@file}}}

\def\includephoto@c{%
  \raisebox{-.5\height}{%
    \includegraphics[width=\photo@width]{\photo@file}}}

\def\includephoto@b{\includegraphics[width=\photo@width]{\photo@file}}

```

And here are the different versions of the `\makeheaders` command:

```
\newlength\leftheader@width
\newlength\rightheader@width

\def\makeheaders@l#1{%
  \setlength\photo@width{\photo@scale\textwidth}
  \setlength\leftheader@width{%
    (\textwidth - \photo@width - \photosep) * \real{\header@scale}}
  \setlength\rightheader@width{%
    \textwidth - \photo@width - \photosep - \leftheader@width}
  \parbox[#1]{\photo@width + \photosep}{\includephoto@\hspace\photosep}%
  \parbox[#1]{\leftheader@width}{\@leftheader}%
  \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}

\def\makeheaders@c#1{%
  \setlength\photo@width{\photo@scale\textwidth}
  \setlength\leftheader@width{(\textwidth - \photo@width) * \real{.5}}
  \setlength\rightheader@width{\leftheader@width}
  \parbox[#1]{\leftheader@width}{\@leftheader}%
  \parbox[#1]{\photo@width}{\includephoto@}%
  \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}

\def\makeheaders@r#1{%
  \setlength\photo@width{\photo@scale\textwidth}
  \setlength\leftheader@width{%
    (\textwidth - \photo@width - \photosep) * \real{\header@scale}}
  \setlength\rightheader@width{%
    \textwidth - \photo@width - \photosep - \leftheader@width}
  \parbox[#1]{\leftheader@width}{\@leftheader}%
  \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}%
  \parbox[#1]{\photo@width + \photosep}{\hspace\photosep\includephoto@}}

\def\makeheaders@#1{%
  \setlength\leftheader@width{\header@scale\textwidth}%
  \setlength\rightheader@width{\textwidth - \leftheader@width}%
  \parbox[#1]{\leftheader@width}{\@leftheader}%
  \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}

\newcommand\makeheaders[1][c]{%
  \strokfalse\strtest{#1}{t}\strtest{#1}{b}\strtest{#1}{c}%
  \ifstrok\else\ClassError{curve}{Invalid argument to \protect\makeheaders}{%
    Argument of \protect\makeheaders must be ‘t’, ‘b’ or ‘c’.}\fi
  \def\tmp@cmd{\global\let\includephoto@}
  \expandafter\tmp@cmd\csname includephoto@#1\endcsname
  \makeheaders@{#1}%
  \par\vspace\headerspace}

```

7.2.2 Titles

\titlefont
\titlespace

```
\@onlypreamble\title

\def\@titlefont{\Huge\bfseries}
\newcommand\titlefont[1]{\gdef\@titlefont{#1}}
\@onlypreamble\titlefont

\newlength\titlespace
\setlength\titlespace{0pt}

```

\subtitle
\subtitlefont

```
\let\@subtitle\@undefined
\newcommand\subtitle[1]{\gdef\@subtitle{#1}}
\@onlypreamble\subtitle

\def\@subtitlefont{\huge\itshape}
\newcommand\subtitlefont[1]{\gdef\@subtitlefont{#1}}
\@onlypreamble\subtitlefont

```

\maketitle

```
\newcommand\maketitle{%
  \begin{center}
    {\@titlefont\@title}
    \ifx\@subtitle\@undefined\else\\\@subtitlefont\@subtitle\fi
  \end{center}
  \vspace\titlespace}

```

7.2.3 Rubric Inclusion

\flavor

```
\let\@flavor\empty
\newcommand\flavor[1]{\gdef\@flavor{#1}
  \ifx\@flavor\empty\else\edef\@flavor{.\@flavor}\fi}

\DeclareOption{ask}{%
  \typein[\@flavor]{Please specify a CV flavor (none by default):}
  \ifx\@flavor\empty\else\edef\@flavor{.\@flavor}\fi}

```

`\input` is redefined in order to deal with flavors and the `ltx` extension.

```
\def\@curveinput#1{%
  \IfFileExists{#1\@flavor.ltx}{\@iinput{#1\@flavor.ltx}}{%
    \IfFileExists{#1\@flavor.tex}{\@iinput{#1\@flavor.tex}}{%
      \IfFileExists{#1.ltx}{\@iinput{#1.ltx}}{%
        \IfFileExists{#1.tex}{\@iinput{#1.tex}}{%
          \@iinput{#1}}}}}}

\renewcommand\input{\@ifnextchar\bgroup\@curveinput\@@input}

```

\makerubric

```
\newcommand\makerubric[1]{\LTXtable{\@almosttextwidth}{#1}}

```

7.2.4 Bibliography

```
\let\newblock\par
\newcounter{bibcount}

```

\bibliography

The following switch is used later in `\bibliography` to shut up a `bibentry` warning.

```
\newif\if@bibentry
\AtBeginDocument{\@ifpackageloaded{bibentry}{\@bibentrytrue}{}}
```

```

\def\bibliography#1{%
  \if@filesw
    \if@bibentry\global\BR@starfalse\fi
    \immediate\write\@auxout{\string\bibdata{#1}}%
  \fi
  \IfFileExists{\jobname.bbl}{\makerubric{\jobname.bbl}}{%
    \typeout{No file \jobname.bbl.}}

```

`\listpubname`

```
\newcommand\listpubname[1]{\gdef\@listpubname{#1}}

```

`thebibliography`

```
\newenvironment{thebibliography}[1]{%
```

The internal bibitem macro definitions are postponed until here for two reasons:

1. First, they have to be delayed until the beginning of the document because `hyperref` redefines them, so we have to override these new definitions yet again.
2. Second, a question of compatibility with `bibentry`: this package defines its own bibliographic environment, in which case the standard bibitem internals should be used.

```
  \def\@lbibitem[##1]##2{\@sentry[\@biblabel{##1}]%
    \if@filesw{%
      \let\protect\noexpand%
      \immediate\write\@auxout{\string\bibcite{##2}{##1}}}
    \fi%
    \ignorespaces}
  \def\@bibitem##1{\@sentry[\stepcounter{bibcount}\@biblabel{\thebibcount}]%
    \if@filesw%
    \immediate\write\@auxout{\string\bibcite{##1}{\thebibcount}}%
    \fi%
    \ignorespaces}
```

`\bibitem` Version 1.8 broke support for standard bibliography because of the `\noalign` trickery. To avoid “misplaced noalign” errors, the solution I found is to issue the `\noalign` as soon as possible, hence the following redefinition of `\bibitem` which actually duplicates the job of `\entry`. Maybe there is a better solution to this problem.

```
  \renewcommand\bibitem{%
    \@nextentry
    \noalign\bgroup\gdef\@beforespace{-\subrubricbeforespace}%
    \@ifnextchar[\@lbibitem\@bibitem}
  \begin{rubric}{\@listpubname}
  }{%
  \end{rubric}
}

```

7.3 Language Processing

```
\DeclareOption{english}{%
```

```
  \continuedname{~(continued)}
  \listpubname{List of Publications}}
\DeclareOption{french}{%
  \continuedname{~(suite)}
  \listpubname{Liste des Publications}}
\DeclareOption{francais}{%
  \ExecuteOptions{french}}
\DeclareOption{spanish}{%
  \continuedname{~(contin\'ua)}
  \listpubname{Lista de Publicaciones}}
\DeclareOption{italian}{%
  \continuedname{~(continua)}
  \listpubname{Pubblicazioni}}
\DeclareOption{german}{%
  \continuedname{~(fortgesetzt)}
  \listpubname{Verzeichnis der Ver\"offentlichungen}}
\DeclareOption{ngerman}{%
  \ExecuteOptions{german}}
\DeclareOption{danish}{%
  \continuedname{~(fortsat)}
  \listpubname{Udgivelser}}
\DeclareOption{dutch}{%
  \continuedname{~(vervolg)}
  \listpubname{Publicaties}}
\DeclareOption{portuges}{%
  \continuedname{~(continua\c c\~ao)}
  \listpubname{Publica\c c\~oes}}
\DeclareOption{portuguese}{%
  \ExecuteOptions{portuges}}
\DeclareOption{brazil}{%
  \ExecuteOptions{portuges}}
\DeclareOption{brazilian}{%
  \ExecuteOptions{portuges}}

```

7.4 Standard Class Processing

```
\DeclareOption{a4paper}{
  \setlength\paperheight{297mm}
  \setlength\paperwidth{210mm}}
\DeclareOption{a5paper}{
  \setlength\paperheight{210mm}
  \setlength\paperwidth{148mm}}
\DeclareOption{b5paper}{
  \setlength\paperheight{250mm}
  \setlength\paperwidth{176mm}}
\DeclareOption{letterpaper}{
  \setlength\paperheight{11in}
  \setlength\paperwidth{8.5in}}
\DeclareOption{legalpaper}{
  \setlength\paperheight{14in}
  \setlength\paperwidth{8.5in}}
\DeclareOption{executivepaper}{
  \setlength\paperheight{10.5in}
  \setlength\paperwidth{7.25in}}
```

```
\DeclareOption{landscape}{
  \setlength\@tempdima{\paperheight}
  \setlength\paperheight{\paperwidth}
  \setlength\paperwidth{\@tempdima}}

\DeclareOption{10pt}{\def\@ptsize{0}}
\DeclareOption{11pt}{\def\@ptsize{1}}
\DeclareOption{12pt}{\def\@ptsize{2}}

\DeclareOption{oneside}{\@twosidefalse\@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue\@mparswitchtrue}

\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\DeclareOption{final}{%
  \setlength\overfullrule{0pt}
  \setlongtables}

\ExecuteOptions{english,letterpaper,10pt,oneside,final}
\ProcessOptions

\input{size1\@ptsize.clo}
\setlength\parindent{0pt}
\setlength\parskip{0pt}
\setlength\tabcolsep{10pt}
\setlength\arrayrulewidth{.4\p@}
\raggedbottom
\onecolumn
\pagestyle{empty}
\pagenumbering{arabic}

\if@twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
    \def\@rubricmark##1{\markboth{\MakeUppercase{##1}}{}}%
    \def\@subrubricmark##1{\markright{##1}}%
  }
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
    \def\@rubricmark##1{\markright{\MakeUppercase{##1}}}%
  }
\fi
\def\ps@myheadings{%
  \let\@oddfoot\@empty\let\@evenfoot\@empty
  \def\@evenhead{\thepage\hfil\slshape\leftmark}%
  \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
  \let\@mkboth\@gobbletwo
  \let\@rubricmark\@gobble
  \let\@subrubricmark\@gobble
}

```

Well, I think that's it. Enjoy using CurVe!